IN THE CLAIMS

Please amend the claims as follows:

Claim 1 (Canceled).

Claim 2 (Currently Amended): An adaptive equalization method in ~~In~~ decoding process for decoding a received signal for a code bit b(n) by iterating adaptive equalization of said received signal for a code bit b(n) a plurality of times, an adaptive equalization ~~method~~ in which a second soft decision value $\lambda_2[b(n)]$ generated by the soft decision of said code bit b(n) is provided as a priori information, and the equalization of a received signal $\mathbf{R}(n)$ of an M-channel ~~channel~~ through utilization of said a priori information and outputting of a first soft decision value $\lambda_1[b(n)]$ are iterated to adaptively equalize said received signal $\mathbf{R}(n)$, said method comprising the steps of:

in a first round of equalization,

(a) calculating an estimated impulse response value $\mathbf{H}_m(n)$ of said M-channel channel and ~~the~~ tap ~~coefficient~~ coefficients $\mathbf{G}(n)$ for linear filtering by the first round of adaptive equalization of said received signal $\mathbf{R}(n)$ of said M-channel, and outputting said first soft decision value $\lambda_1[b(n)]$;

in the second and subsequent rounds of equalization,

(b) calculating the likelihood b′(n) of a code bit sequence b(n) from said second soft decision value $\lambda_2[b(n)]$ obtained by decoding said code bit b(n) based on said first soft decision value $\lambda_1[b(n)]$;

(c) calculating an estimated impulse response value vector $\mathbf{H}_L(n)$ by ~~approximated~~ approximation with the intersymbol interference component for the code bit b(n) regarded as zero, and calculating the tap ~~coefficient~~ coefficients $\mathbf{G}(n)$ for said linear filtering from said

estimated impulse response value vector $\mathbf{H}_L(n)$ to update said tap ~~coefficient~~ coefficients $\mathbf{G}(n)$;

(d) linear-filtering said likelihood b′(n) of said code bit sequence b(n) with said estimated impulse response value $\mathbf{H}_L(n)$ to generate a replica of the received signal $\mathbf{R}(n)$;

(e) subtracting said replica from said received signal $\mathbf{R}(n)$ to generate a difference signal $\mathbf{R}_c(n)$ without intersymbol interference;

(f) linear-filtering said difference signal $\mathbf{R}_c(n)$ with said tap ~~coefficient~~ coefficients $\mathbf{G}(n)$ to generate a signal $\mathbf{Z}(n)$; and

(g) outputting, as the results of said second and subsequent rounds of adaptive equalization, the ~~a~~ first soft decision value $\lambda_1[b(n)]$ updated with said signal $\mathbf{Z}(n)$ and said estimated impulse response value vector $\mathbf{H}_L(n)$.

Claim 3 (Currently Amended): The method of claim 2, wherein said step (a) comprises the steps of:

(a-1) calculating the estimated impulse response value $\mathbf{H}_m(n)$ of said M-channel channel based on said received signal $\mathbf{R}(n)$ and a known signal;

(a-2) calculating said tap ~~coefficient~~ coefficients $\mathbf{G}(n)$ by an adaptive algorithm based on said received signal $\mathbf{R}(n)$ and said known signal;

(a-3) linear-filtering said received signal $\mathbf{R}(n)$ with said tap ~~coefficient~~ coefficients $\mathbf{G}(n)$; and

(a-4) calculating said first soft decision value $\lambda_1[b(n)]$ from said signal $\mathbf{Z}(n)$ and said estimated impulse response value $\mathbf{H}_m(n)$.

Claim 4 (Currently Amended): The method of claim 2 or 3, wherein said step (c) includes a step of calculating the tap ~~coefficient~~ coefficients $\mathbf{G}(n)$ for said linear filtering

from said estimated impulse response value vector $\mathbf{H}_L(n)$ through utilization of a Matrix Inversion Lemma.~~,~~

Claim 5 (Currently Amended): The method of claim 2, wherein said step (a) comprises the steps of:

(a-1) calculating the estimated impulse response value $\mathbf{H}_m(n)$ of said M-channel channel based on a sample value sequence of said received signal $\mathbf{R}(n)$ and a known signal;

(a-2) determining from said estimated impulse response value $\mathbf{H}_m(n)$ whether the received signal power is larger than a predetermined reference value, and, if larger than said reference value, storing the corresponding path as an effective path in a memory;

(a-3) calculating said tap ~~coefficient~~ coefficients $\mathbf{G}(n)$ based on said received signal $\mathbf{R}(n)$ and said known signal, storing a tap ~~coefficient~~ coefficients $\mathbf{G}'(n)$ for said linear filtering corresponding to said effective path in a memory, and storing a received signal vector $\mathbf{R}'(n)$ corresponding to said effective path in a memory; and

(a-4) calculating said first soft decision value $\lambda_1[b(n)]$ from said estimated impulse response value $\mathbf{H}_m(n)$, said tap ~~coefficient~~ coefficients $\mathbf{G}'(n)$ and said received signal vector $\mathbf{R}'(n)$.

Claim 6 (Currently Amended): The method of claim 5, wherein:

said step (c) comprises the steps of:

(c-1) calculating the tap ~~coefficient~~ coefficients $\mathbf{G}'(n)$ for said linear filtering corresponding to said effective path from said estimated impulse response value $\mathbf{H}_L'(n)$ composed of the ~~component~~ components corresponding to said effective path through utilization of an Matrix Inversion Lemma; and

(c-2) storing said tap ~~coefficient~~ coefficients $\mathbf{G}'(n)$ for said linear filtering corresponding to said effective path in a memory;

said step (d) is a step of linear-filtering said tap ~~coefficient~~ coefficients $\mathbf{G}'(n)$ with said estimated impulse response value vector $\mathbf{H}_L'(n)$ to obtain a replica signal;

said step (e) is a step of storing in a memory a difference signal $\mathbf{R}_c'(n)$ corresponding to said effective path, said difference signal $\mathbf{R}_c'(n)$ being obtained by subtracting said replica signal from said received signal $\mathbf{R}'(n)$;

said step (f) is a step of linear-filtering said difference signal $\mathbf{R}_c'(n)$ with said ~~r~~tap ~~coefficient~~ coefficients $\mathbf{G}'(n)$ to generate a signal $\mathbf{Z}'(n)$; and

said step (g) is a step of calculating said first soft decision value $\lambda_1[b(n)]$ from said estimated impulse response value vector $\mathbf{H}_L'(n)$ and said signal $\mathbf{Z}'(n)$.

Claim 7 (Currently Amended): The method of claim 2, wherein, letting J represent the maximum number of delayed symbols to be considered, and letting a received signal sample vector of an M-channel channel be represented by $\mathbf{r}(n)=[r_0(n)r_1(n)\ldots r_{M-1}(n)]^T$, said received signal vector $\mathbf{R}(n)$ by $\mathbf{R}(n)=[\mathbf{r}(n+J-1)\mathbf{r}(n+J-2)\ldots\mathbf{r}(n)]^T$, a channel weighting coefficient vector by $\mathbf{h}(n;j)=[h_0(n;j)h_1(n;j)\ldots h_{M-1}(n;j)]^T$, and a channel matrix $\mathbf{H}_m(n)$ of said estimated impulse response value by

$$\mathrm{H}_m(n)=\begin{bmatrix} \mathbf{h}(n;0) & \mathbf{h}(n;1) & \cdots & \mathbf{h}(n;J-1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{h}(n;0) & \mathbf{h}(n;1) & \cdots & \mathbf{h}(n;J-1) & \mathbf{0} & \mathbf{0} \\ \vdots & & & \ddots & & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{h}(n;0) & \mathbf{h}(n;1) & \cdots & \mathbf{h}(n;J-1) & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{h}(n;0) & \mathbf{h}(n;1) & \cdots & \mathbf{h}(n;J-1) \end{bmatrix}_{MJ\times(2J-1)}$$

said step (a) comprises the steps of:

(a-1) linear-filtering a training signal b(n) in a training signal period with said estimated impulse response value $\mathbf{H}_m(n)$ to generate a replica $\mathbf{H}_m(n)b(n)$;

(a-2) generating the difference between said received signal $\mathbf{R}(n)$ and said replica $\mathbf{H}_m(n)b(n)$ as a difference vector $\mathbf{R}_c(n)$;

(a-3) linear-filtering said received signal $\mathbf{R}(n)$ with said tap ~~coefficient~~ coefficients $\mathbf{G}(n)$ to generate an output $\mathbf{Z}(n)=\mathbf{G}(n)^H\mathbf{R}(n)$;

(a-4) determining said tap ~~coefficient~~ coefficients $\mathbf{G}(n)$ by an adaptive algorithm based on the difference between said output $\mathbf{Z}(n)$ and said training signal $b(n)$; and

(a-5) calculating a soft decision value $\lambda_1[b(n)]=4\text{Real}\{\mathbf{Z}(n)\}/(1-\mu)$ based on the estimated impulse response value $\mathbf{H}_m(n)$ for linear filtering and said output $\mathbf{Z}(n)$, and outputting said soft decision value $\lambda_1[b(n)]=4\text{Real}\{\mathbf{Z}(n)\}/(1-\mu)$ as the result of said first round of equalization;

said step (b) is a step of calculating the likelihood $b'(k)=\tanh[\lambda_2[b(k)/2]$ of a code bit sequence $b(k)$ from a soft decision value $\lambda_2[b(n)]$ of a decoded bit provided as a priori information with said k set within the range of $n-(j-1)\leq k\leq n+(J-1)$;

said step (c) is a step of calculating said tap ~~coefficient~~ coefficients $\mathbf{G}(n)$ by approximating

$$G(n)=[H_L(n)H_L(n)^H(n)-\sigma^2 I]^{-1}H_L(n)$$

$$H_L(n)=[h_0(n;J-1)\ldots h_{M-1}(n;J-1)h_0(n;J-2)\ldots h_{M-1}(n;J-2)\ldots h_0(n;0)\ldots h_{M-1}(n;0)]^T$$

said step (d) is a step of linear-filtering, with said $\mathbf{H}_L(n)$, an estimated value vector,

$$\mathbf{B}'(n)=[b'(n+(J-1))b'(n+(J-2))\ldots b'(n+1)0b'(n-1)\ldots b'(n-(J-1))]^T,$$

of a code bit that affects, as intersymbol interference, said code bit $b(n)$ at time n to thereby obtain a replica $\mathbf{H}_L(n)\mathbf{B}'(n)$;

said step (e) is a step of calculating a difference vector $\mathbf{R}_c(n)=\mathbf{R}(n)-\mathbf{H}_L(n)\mathbf{B}'(n)$ between said replica $\mathbf{H}_L(n)\mathbf{B}'(n)$ and said received signal $\mathbf{R}(n)$;

said step (f) is a step of linear-filtering said difference vector $\mathbf{R}_c(n)$ with said tap ~~coefficient~~ coefficients $\mathbf{G}(n)$ and outputting the result of said linear filtering $\mathbf{Z}(n)=\mathbf{G}(n)^H\mathbf{R}_c(n)$; and

said step (g) is a step of obtaining a soft decision value

$$\lambda_1[b(n)] = \frac{4\mathrm{Real}\{Z(n)\}}{1-\mu(n)}$$

$\mu(n)=H_L(n)^H G(n)$

as the output of said second and subsequent rounds of equalization from said output $\mathbf{Z}(n)$ and said estimated impulse response value vector $\mathbf{H}_L(n)$.

Claim 8 (Currently Amended): The method of claim 6, wherein, letting J represent the maximum number of delayed symbols to be considered, and letting a received signal sample vector of an M-channel channel be represented by $\mathbf{r}(n)=[r_0(n)r_1(n)\ldots r_{M-1}(n)]^T$, said received signal vector $\mathbf{R}(n)$ by $\mathbf{R}(n)=[\mathbf{r}(n+J-1)\mathbf{r}(n+J-2)\ldots\mathbf{r}(n)]^T$, a channel weighting coefficient vector by $\mathbf{h}(n;j)=[h_0(n;j)h_1(n;j)\ldots h_{M-1}(n;j)]^T$, and a channel matrix $\mathbf{H}_m(n)$ of said estimated impulse response value by

$$H_m(n) = \begin{bmatrix} \mathbf{h}(n;0) & \mathbf{h}(n;1) & \cdots & \mathbf{h}(n;J-1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{h}(n;0) & \mathbf{h}(n;1) & \cdots & \mathbf{h}(n;J-1) & \mathbf{0} & \mathbf{0} \\ \vdots & & & \ddots & & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{h}(n;0) & \mathbf{h}(n;1) & \cdots & \mathbf{h}(n;J-1) & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{h}(n;0) & \mathbf{h}(n;1) & \cdots & \mathbf{h}(n;J-1) \end{bmatrix}_{MJ\times(2J-1)}$$

said step (a-3) comprises the steps of:

(a-3-1) linear filtering a training signal b(n) in a training signal period with said estimated impulse response value $\mathbf{H}_m(n)$ to generate a replica $\mathbf{H}_m(n)b(n)$;

(a-3-2) generating the difference between said received signal $\mathbf{R}(n)$ and said replica $\mathbf{H}_m(n)b(n)$ as a difference vector $\mathbf{R}_c(n)$;

(a-3-3) linear filtering said received signal **R**(n) with said tap ~~coefficient~~ coefficients **G**(n) to generate an output $\mathbf{Z}(n)=\mathbf{G}(n)^H\mathbf{R}(n)$; and

(a-3-4) determining said tap ~~coefficient~~ coefficients **G**(n) by an adaptive algorithm based on the difference between said output **Z**(n) and said training signal b(n), and storing a received signal **R**′(n) and a tap ~~coefficient~~ coefficients **G**′(n) of those components of said received signal **R**(n) and said tap ~~coefficient~~ coefficients **G**(n) which correspond to said effective path;

said step (a-4) is a step of calculating a soft decision value $\lambda_1[b(n)]=4\mathrm{Real}\{\mathbf{Z}(n)\}/(1-\mu)$ based on the estimated impulse response value $\mathbf{H}_m'(n)$ for said linear filtering and said output **Z**(n), and outputting said soft decision value $\lambda_1[b(n)]=4\mathrm{Real}\{\mathbf{Z}(n)\}/(1-\mu)$ as the result of said first round of equalization;

said step (b) is a step of calculating the likelihood $b'(k)=\tanh[\lambda_2[b(k)/2]$ of a code bit sequence b(k) from a soft decision value $\lambda_2[b(n)]$ of a decoded bit provided as a priori information with said k set within the range of $n-(j-1)\le k\le n+(J-1)$;

said step (c-1) is a step of calculating said tap ~~coefficient~~ coefficients **G**′(n) by approximating

$$G'(n)=[H_L'(n)H_L'(n)^H(n)-\sigma^2 I]^{-1}H_L'(n)$$

$$H_L'(n)=[h_0(n;J-1)\ldots h_{M-1}(n;J-1)h_0(n;J-2)\ldots h_{M-1}(n;J-2)\ldots h_0(n;0)\ldots h_{M-1}(n;0)]^T$$

said step (d) is a step of linear filtering, with said HL′(n), an estimated value vector,

$$\mathbf{B}'(n)=[b'(n+(J-1))b'(n+(J-2))\ldots b'(n+1)0b'(n-1)\ldots b'(n-(J-1))]^T,$$

of a code bit that affects, as intersymbol interference, said code bit b(n) at time n to thereby obtain a replica $\mathbf{H}_L(n)\mathbf{B}'(n)$;

said step (e) is a step of calculating a difference vector $\mathbf{R}_c'(n)=\mathbf{R}'(n)-\mathbf{H}_L'(n)\mathbf{B}'(n)$ between said replica $\mathbf{H}_L'(n)\mathbf{B}'(n)$ and said received signal **R**′(n);

said step (f) is a step of linear-filtering said difference vector $\mathbf{R}_c'(n)$ with said tap ~~coefficient~~ coefficients $\mathbf{G}'(n)$ and outputting the result of said linear filtering $\mathbf{Z}'(n)=\mathbf{G}'(n)^H\mathbf{R}_c'(n)$; and

said step (g) is a step of obtaining a soft decision value

$$\lambda_1[b(n)] = \frac{4\mathrm{Real}\{Z'(n)\}}{1-\mu(n)}$$

$\mu(n)=H_L'(n)^H G'(n)$

as the output of said second and subsequent rounds of equalization from said output $\mathbf{Z}'(n)$ and said estimated impulse response value vector $\mathbf{H}_L'(n)$.

Claim 9 (Currently Amended): An adaptive equalizer comprising:

an impulse response estimating part for calculating an estimated impulse response value $\mathbf{H}_m(n)$ of each of Channel based on a received signal $\mathbf{R}(n)$ and a known signal;

a tap ~~coefficient~~ coefficients calculating part for calculating ~~the~~ tap ~~coefficient~~ coefficients $\mathbf{G}(n)$ of a linear filter by an adaptive algorithm based on said received signal $\mathbf{R}(n)$ and said known signal;

said linear filter having set therein said tap ~~coefficient~~ coefficients $\mathbf{G}(n)$, for linear-filtering said received signal $\mathbf{R}(n)$; ~~and~~

a soft decision value calculating part for calculating a soft decision value $\Lambda_1[b(n)]$ from said estimated impulse response value $\mathbf{H}_m(n)$ and the result of said linear filtering;

a storage part for storing said estimated impulse response value $\mathbf{H}_m(n)$;

a likelihood calculating part for calculating, from said soft decision value $\Lambda_2[b(n)]$, its likelihood b'(n); and

means for obtaining a replica of said received signal by subjecting said likelihood b'(n) to linear filtering with an estimated intersymbol interference vector $\mathbf{B}'(n)$ obtained from

said stored estimated impulse response value vector $\mathbf{H}_m(n)$ by approximating intersymbol interference components with respect to the code bit b(n) to zero;

wherein said tap coefficients calculating part includes means for calculating the tap coefficients **G**(n) of said linear filter by an adaptive algorithm based on said received signal **R**(n) and said known signal when said ~~a~~ soft decision value is not input to said tap coefficients calculating part, and for calculating said tap coefficients **G**(n) from said estimated impulse response value vector $\mathbf{H}_m(n)$ when said soft decision value is input to said tap coefficients calculating part; and

said linear filter being adapted to performs linear filtering of said received signal **R**(n), in the absence of said soft decision value, and the difference between said received signal and said replica signal $\mathbf{H}_m(n)\mathbf{B}'(n)$, in the presence of said soft decision value through the use of said tap coefficients **G**(n), and provides the linear filtering output to said soft decision value calculating part.

Claim 10 (Canceled).

Claim 11 (Currently Amended): The adaptive equalizer of claim 9, further comprising:

a path decision part for determining from said estimated impulse response value $\mathbf{H}_m(n)$ of said each channel whether the received power of the corresponding path is larger than a predetermined reference value h_{th};~~;~~

a path memory for storing as an effective path a path determined as being larger than said predetermined reference value h_{th};

a tap ~~coefficient~~ coefficients memory for storing as ~~a~~ new tap ~~coefficient~~ coefficients **G'(n)** the ~~component~~ components of that one of said tap coefficients corresponding to said effective path; and

a signal identifying part for identifying that received signal ~~component~~ components of said received signal **R(n)** corresponding to said effective path;

wherein said soft decision value calculating part calculates said soft decision value from the received signal ~~component~~ components corresponding to said effective path, the estimated impulse response value corresponding to said effective path and the tap ~~coefficient~~ coefficients corresponding to said effective path.

Claim 12 (Currently Amended): The adaptive equalizer of claim 11, further comprising:

a likelihood calculating part for calculating the likelihood b'(n) of a code from said soft decision value;

a replica generating linear filter for generating a replica signal $\mathbf{H}_L(n)b'(n)$ of said received signal by linear-filtering said likelihood b'(n) with an estimated impulse response value $\mathbf{H}_L(n)$ composed of that ~~component~~ components of said estimated response value vector corresponding to said effective path;

a subtractor for subtracting said replica signal $\mathbf{H}_L(n)b'(n)$ from that ~~component~~ components **R'(n)** of said received signal corresponding to said effective path to obtain a difference signal; and

~~a difference memory for storing only that difference signal component of said difference signal corresponding to said effective path;~~

wherein said tap ~~coefficient~~ coefficients calculating part calculates, in second and subsequent rounds of equalization, the tap ~~coefficient~~ coefficients **G'(n)** of said linear filter

from said estimated impulse response value vector $\underline{\mathbf{H}_L'(n)}$ corresponding to said effective path through the use of a Matrix Inversion Lemma; and

said soft decision calculating part is a means for calculating, in said second and subsequent rounds of equalization, said soft decision value $\underline{\Lambda_1[b(n)]}$ from said estimated impulse response value vector $\underline{\mathbf{H}_L'(n)}$ corresponding to said effective path, said tap ~~coefficient~~ coefficients $\underline{\mathbf{G}'(n)}$ of said linear filter corresponding to said effective path and said difference signal $\underline{\mathbf{R}_c'(n)}$ corresponding to said effective path.